

NNN	NNN	MMM	MMM	LLL
NNN	NNN	MMM	MMM	LLL
NNN	NNN	MMM	MMM	LLL
NNN	NNN	MMMMMM	MMMMMM	LLL
NNN	NNN	MMMMMM	MMMMMM	LLL
NNN	NNN	MMMMMM	MMMMMM	LLL
NNNNNN	NNN	MM	MM	LLL
NNNNNN	NNN	MM	MM	LLL
NNNNNN	NNN	MM	MM	LLL
NN NNN	NNN	MM	MM	LLL
NN NNN	NNN	MM	MM	LLL
NN NNN	NNN	MM	MM	LLL
NNN	NNNNNN	MM	MM	LLL
NNN	NNNNNN	MM	MM	LLL
NNN	NNNNNN	MM	MM	LLL
NNN	NNN	MM	MM	LLL
NNN	NNN	MM	MM	LLL
NNN	NNN	MM	MM	LLL
NNN	NNN	MM	LLLLLLLLLLLL	
NNN	NNN	MM	LLLLLLLLLLLL	
NNN	NNN	MM	LLLLLLLLLLLL	

FILEID**NMLPMANIP

N 11

NN NN MM MM LL P PPPPPP MM MM AAAA NN NN IIII P PPPPPPPP
 NN NN MM MM LL P PPPPPP MM MM AAAA NN NN IIII P PPPPPPPP
 NN NN MMMM MMMM LL PP PP MMMM MMMM AA AA NN NN II PP PP
 NN NN MMMM MMMM LL PP PP MMMM MMMM AA AA NN NN II PP PP
 NNNN NN MM MM MM LL PP PP MM MM MM AA AA NNNN NN NN II PP PP
 NNNN NN MM MM MM LL PP PP MM MM MM AA AA NNNN NN NN II PP PP
 NN NN NN MM MM LL PPPPPP MM MM AA AA NN NN NN NN III PPPPPP
 NN NN NN MM MM LL PPPPPP MM MM AA AA NN NN NN NN III PPPPPP
 NN NNNN MM MM LL PP MM MM AAAA NN NNNN NN II PP
 NN NNNN MM MM LL PP MM MM AAAA NN NNNN NN II PP
 NN NN MM MM LL PP MM MM AA AA NN NN NN NN II PP
 NN NN MM MM LL PP MM MM AA AA NN NN NN NN II PP
 NN NN MM MM LLLLLLLL PP MM MM AA AA NN NN IIII PP
 NN NN MM MM LLLLLLLL PP MM MM AA AA NN NN IIII PP

```
1      0 XTITLE 'NML internal parameter manipulation module'
2      0 MODULE NML$PMANIP (
3          LANGUAGE (BLISS32),
4          ADDRESSING_MODE (NONEXTERNAL=GENERAL),
5          ADDRESSING_MODE (EXTERNAL=GENERAL),
6          IDENT = 'V04-000'
7          ) =
8      1 BEGIN
9
10     1 ****
11    1 *
12    1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13    1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14    1 * ALL RIGHTS RESERVED.
15    1 *
16    1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17    1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18    1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19    1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20    1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21    1 * TRANSFERRED.
22    1 *
23    1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24    1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25    1 * CORPORATION.
26    1 *
27    1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28    1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29    1 *
30    1 *
31    1 ****
32    1 *
33    1 *
34    1 ABSTRACT:
35    1 *
36    1 This module contains routines to handle internal parameter
37    1 manipulation functions.
38    1 *
39    1 ENVIRONMENT: VAX/VMS Operating System
40    1 *
41    1 AUTHOR: Distributed Systems Software Engineering
42    1 *
43    1 CREATION DATE: 23-JAN-1980
44    1 *
45    1 MODIFIED BY:
46    1 *
47    1   V03-003 MKP0003 Kathy Perko 4-Aug-1983
48    1   Make permanent database routines transparent to the length
49    1   of the ISAM keys at the beginning of the records.
50    1 *
51    1   V03-002 MKP0002 Kathy Perko 22-June-1982
52    1   Add support for specifying "active X25-Protocol network".
53    1 *
54    1   V03-001 MKP0001 Kathy Perko 28-April-1982
55    1   More modifications for NETACP control QIO. Add the
56    1   second search key to NFB. Also, delete the start key.
57    1 *
```

NMLSPMANIP
V04-000

NML internal parameter manipulation module

C 12
16-Sep-1984 00:26:09 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:16 [NML.SRC]NMLPMANIP.B32;1

Page 2
(1)

: 58
: 59
: 60
: 61

0058 1 ! V02-001 LMK0001 Len Kawell 21-Jul-1981
0059 1 !-- Modifications for new NETACP control QIO.
0060 1 !--
0061 1 !--

NM
VO

```
: 63      0062 1 %SBTTL 'Declarations'  
.: 64      0063 1  
.: 65      0064 1  
.: 66      0065 1 | TABLE OF CONTENTS:  
.: 67      0066 1 |  
.: 68      0067 1 |  
.: 69      0068 1 FORWARD ROUTINE  
.: 70      0069 1 NML$SAVEPARAM,  
.: 71      0070 1 NML$CHKPRMVAL,  
.: 72      0071 1 NML$BLDSETQBF : NOVALUE,  
.: 73      0072 1 NML$DEL_FIELDS,  
.: 74      0073 1 NML$ADD_FIELDS,  
.: 75      0074 1 NML$READPARLIST,  
.: 76      0075 1 NML$SHOWPARLIST,  
.: 77      0076 1 NML$BLDALLDIS;  
.: 78      0077 1  
.: 79      0078 1 |  
.: 80      0079 1 | INCLUDE FILES:  
.: 81      0080 1 |  
.: 82      0081 1 |  
.: 83      0082 1 LIBRARY 'LIB$:NMLLIB.L32';  
.: 84      0083 1 LIBRARY 'SHRLIB$:NMALIBRY.L32';  
.: 85      0084 1 LIBRARY 'SHRLIB$:NET.L32';  
.: 86      0085 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';  
.: 87      0086 1  
.: 88      0087 1 |  
.: 89      0088 1 | EXTERNAL REFERENCES:  
.: 90      0089 1 |  
.: 91      0090 1 |  
.: 92      0091 1 $NML_EXTDEF:  
.: 93      0092 1  
.: 94      0093 1 EXTERNAL LITERAL  
.: 95      0094 1 NMLS_DSCBFOVF,  
.: 96      0095 1 NMLS_QIOBFOVF,  
.: 97      0096 1 NMLS_RECVFOVF;  
.: 98      0097 1  
.: 99      0098 1 EXTERNAL ROUTINE  
.: 100     0099 1 NMASSEARCHFLD,  
.: 101     0100 1 NML$BLD_REPLY,  
.: 102     0101 1 NML$BLDP2,  
.: 103     0102 1 NML$ERROR_1,  
.: 104     0103 1 NML$ERROR_2;  
.: 105     0104 1
```

```
: 107      0105 1 %SBTTL 'NML$SAVEPARAM Check parameter value'
108      0106 1 GLOBAL ROUTINE NML$SAVEPARAM (CPT_INDEX, LENGTH, POINTER) =
109      0107 1
110      0108 1 ++ FUNCTIONAL DESCRIPTION:
111      0109 1
112      0110 1 This routine saves a parameter as a descriptor in the parameter
113      0111 1 descriptor block.
114      0112 1
115      0113 1
116      0114 1 FORMAL PARAMETERS:
117      0115 1
118      0116 1     CPT INDEX
119      0117 1     LENGTH
120      0118 1     POINTER
121      0119 1
122      0120 1
123      0121 1
124      0122 1     NML$AB_PRMSEM is the parameter semantic table.
125      0123 1     NML$AW_PRM_DES is the parameter descriptor buffer.
126      0124 1     NML$GW_PRMDESCNT contains the current number of descriptor entries.
127      0125 1
128      0126 1     IMPLICIT OUTPUTS:
129      0127 1
130      0128 1     If the parameter is valid then a descriptor entry will be created for
131      0129 1     it in NML$AW_PRM_DES and NML$GW_PRMDESCNT will be incremented.
132      0130 1
133      0131 1     ROUTINE VALUE:
134      0132 1     COMPLETION CODES:
135      0133 1
136      0134 1     Always returns NMLS_STS_SUC.
137      0135 1
138      0136 1
139      0137 1
140      0138 1     SIDE EFFECTS:
141      0139 1
142      0140 1
143      0141 1
144      0142 1
145      0143 2     BEGIN
146      0144 2
147      0145 2     BIND
148      0146 2     CPT_LIST      = NML$AB_CPTABLE [.CPT_INDEX, 0,0,0,0]
149      0147 2     : BBLOCK [CPT$K_ENTRYLEN]
150      0148 2     SEMANTIC_LIST = NML$AB_PRMSEM [-CPT_LIST [CP1$W_PSTINDEX], 0,0,0,0]
151      0149 2     : BBLOCK [PST$K_ENTRYLEN];
152      0150 2
153      0151 2     LOCAL
154      0152 2     VEC_INDEX,
155      0153 2     MASK      : BLOCK [1, WORD],
156      0154 2     MSGSIZE,           | Resultant message size
157      0155 2     OFFSET,            | Temporary parameter offset
158      0156 2     VEC      : REF BLOCKVECTOR [. 2, WORD];
159      0157 2
160      0158 2
161      0159 2
162      0160 2     | Check the parameter descriptor buffer to see if there is any room left
163      0161 2
```

```

: 164      0162 2   IF .NML$GW_PRMDESCNT GEQU PDBSK_NUMBER
165      0163 2   THEN
166      0164 3   BEGIN
167      0165 3
168      0166 3   | Signal parameter descriptor buffer overflow.
169      0167 3
170      0168 3   NML$AB_MSGBLOCK [MSBSL_FLAGS] = MSBSM_DET_FLD OR MSBSM_MSG_FLD;
171      0169 3   NML$AB_MSGBLOCK [MSBSB_CODE] = NMASC_STS_MPR; ! Get error code
172      0170 3   NML$AB_MSGBLOCK [MSBSW_DETAIL] =
173      0171 3   | SEMANTIC_LIST [PSTS_WDATAID]; ! Get parameter code detail
174      0172 3   NML$AB_MSGBLOCK [MSBSL_TEXT] = NMLS_DSCBF0VF;
175      0173 3   NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE); ! Build message
176      0174 3   $SIGNAL_MSG (NML$AB_SNDBUFFER, .MSGSIZE); ! Signal error message
177      0175 3
178      0176 2   END;
179      0177 2
180      0178 2   | Add descriptor entry for this parameter.
181      0179 2
182      0180 2   NML$AW_PRM_DES [.NML$GW_PRMDESCNT, PDBSW_INDEX] = .CPT_INDEX;
183      0181 2   NML$AW_PRM_DES [.NML$GW_PRMDESCNT, PDBSW_COUNT] = .LENGTH;
184      0182 2   NML$AW_PRM_DES [.NML$GW_PRMDESCNT, PDBSA_POINTER] = .POINTER;
185      0183 2
186      0184 2   NML$GW_PRMDESCNT = .NML$GW_PRMDESCNT + 1; ! Increment descriptor count
187      0185 2
188      0186 2   RETURN NMLS_STS_SUC
189      0187 2
190      0188 1   END;
                           ! End of NML$SAVEPARAM

```

```

.TITLE NML$PMANIP NML internal parameter manipulation
module
.IDENT \V04-000\
```

```

.EXTRN NML$GB_EVTSRCTYP
.EXTRN NML$GQ_EVTSRCDS
.EXTRN NML$GW_EVTCLASS
.EXTRN NML$GB_EVTMSKTY
.EXTRN NML$GQ_EVTMSKDSC
.EXTRN NML$GW_EVTSNKADR
.EXTRN NML$GW_ACP_CHAN
.EXTRN NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
.EXTRN NML$AB_QIOBUFFER
.EXTRN NML$GQ_QIOBFDSC
.EXTRN NML$AB_EXEBUFFER
.EXTRN NML$GL_EXEDATPTR
.EXTRN NML$GQ_EXEDATDSC
.EXTRN NML$GQ_EXEBFDSC
.EXTRN NML$AB_RCVBUFFER
.EXTRN NML$GQ_RCVBFDSC
.EXTRN NML$AB_SNDBUFFER
.EXTRN NML$GQ_SNDBFDSC
.EXTRN NML$GL_RCVDATLEN
.EXTRN NML$AB_CPTABLE, NML$AB_MSGBLOCK
.EXTRN NML$AB_ENTITY_ID
.EXTRN NML$AB_QUALIFIER_ID
.EXTRN NML$AB_ENTITYDATA
.EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM
```

• EXTRN NML\$AB_RECVBUF, NML\$AL_ENTINFTAB
• EXTRN NML\$AL_PERMINFTAB
• EXTRN NML\$AW_PRMDES, NML\$GB_CMD_VER
• EXTRN NML\$GB_ENTITY_CODE
• EXTRN NML\$GB_ENTITY_FORMAT
• EXTRN NML\$GL_QUALIFIER_PST
• EXTRN NML\$GB_QUALIFIER_FORMAT
• EXTRN NML\$GB_FUNCTION
• EXTRN NML\$GB_INFO, NML\$GB_OPTIONS
• EXTRN NML\$GL_PRMCODE, NML\$GL_PRS_FLGS
• EXTRN NML\$GL_NML_ENTITY
• EXTRN NML\$GQ_NETRAMDSC
• EXTRN NML\$GQ_RECVFDSC
• EXTRN NML\$GW_PRMDESCNT
• EXTRN NML\$DSCBFOVF, NML\$QIOBFOVF
• EXTRN NML\$RECBFOVF, NMASSEARCHFLD
• EXTRN NML\$BLD_REPLY, NML\$BLDP2
• EXTRN NML\$ERROR_1, NML\$ERROR_2

PSECT	SCODE\$, NOWRT, 2	
.ENTRY	NMLSSAVEPARAM, Save R2,R3,R4	0106
MOVAB	NMLSAW_PRM_DES, R4	
MOVAB	NMLSGW_PRMDESCNT, R3	
MOVAB	NML\$AB_MSGBLOCK, R2	
SUBL2	#4, SP	
MULL3	#10, CPT INDEX, R0	0146
PUSHAB	NML\$AB_CPTABLE[R0]	0148
MOVZWL	a(SP)+, R0	
MULL2	#16, R0	
CMPW	NMLSGW_PRMDESCNT, #32	0162
BLSSU	1\$	
MOVL	#6, NML\$AB_MSGBLOCK	0168
MNEG B	#5, NML\$AB_MSGBLOCK+4	0169
PUSHAB	NML\$AB_PRMSEM[R0]	0171
MOVW	a(SP)+, NML\$AB_MSGBLOCK+8	
MOVL	#NMLS_DSCBF0VF, NML\$AB_MSGBLOCK+12	0172
PUSHR	#^M<R2,SP>	0173
CALLS	#2, NML\$BLD_REPLY	
PUSHL	MSGSIZE	0174
PUSHAB	NML\$AB_SNDBUFFER	
PUSHL	#33095680	
CALLS	#3, LIB\$SIGNAL	
MOVZWL	NMLSGW_PRMDESCNT, R0	0180
PUSHAQ	NMLSAW_PRM_DES[R0]	
MOVW	CPT INDEX, a(SP)+	
PUSHAQ	NMLSAW_PRM_DES+2[R0]	0181
MOVW	LENGTH, a(SP)+	
PUSHAQ	NMLSAW_PRM_DES+4[R0]	0182
MOVL	POINTER, a(SP)+	
INCW	NMLSGW_PRMDESCNT	0184
MOVL	#1, R0	0186
RET		0188

; Routine Size: 139 bytes, Routine Base: \$CODE\$ + 0000

```
192    0189 1 %SBTTL 'NML$CHKPRMVAL Check parameter value'
193    0190 1 GLOBAL ROUTINE NMLSCHKPRMVAL (CPT_INDEX, LEN, ADR) =
194    0191 1
195    0192 1 ++
196    0193 1 FUNCTIONAL DESCRIPTION:
197    0194 1
198    0195 1 This routine verifies that parameter values from the NICE message
199    0196 1 fall within valid boundaries.
200    0197 1
201    0198 1 FORMAL PARAMETERS:
202    0199 1
203    0200 1     CPT_INDEX      Index into change parameter table.
204    0201 1     LEN            Byte count of parameter.
205    0202 1     ADR            Address of parameter.
206    0203 1
207    0204 1 IMPLICIT INPUTS:
208    0205 1     NONE
209    0206 1
210    0207 1 IMPLICIT OUTPUTS:
211    0208 1     NONE
212    0209 1
213    0210 1 ROUTINE VALUE:
214    0211 1 COMPLETION CODES:
215    0212 1     Returns success (NMLS_STS_SUC) if the parameter value is within range.
216    0213 1
217    0214 1 SIDE EFFECTS:
218    0215 1     An error message (NMA$C_STS_PVA) is signalled if the value is bad.
219    0216 1
220    0217 1
221    0218 1
222    0219 1
223    0220 1
224    0221 1 --
225    0222 1
226    0223 2 BEGIN
227    0224 2
228    0225 2 LOCAL
229    0226 2     MAX,           ! Maximum parameter value (0 if no limit)
230    0227 2     MIN,           ! Minimum parameter value
231    0228 2     VAL,            ! Parameter value to compare
232    0229 2     STATUS;         ! Status of the range checking operations
233    0230 2
234    0231 2 The parameter semantic table index is determined by looking in the change
235    0232 2 parameter table.
236    0233 2
237    0234 2 BIND
238    0235 2     CPT_LIST      = NMLSAB_CPTABLE [.CPT_INDEX, 0,0,0,0]
239    0236 2             : BBLOCK [CPT$K_ENTRYLEN]
240    0237 2     SEMANTIC_LIST = NMLSAB_PRMSEM [CPT_LIST [CPT$W_PSTINDEX], 0,0,0,0]
241    0238 2             : BBLOCK [PST$K_ENTRYLEN];
242    0239 2
243    0240 2 Pick up the values for comparison.
244    0241 2
245    0242 2     MIN = .SEMANTIC_LIST [PST$L_MINVALUE];
246    0243 2     MAX = .SEMANTIC_LIST [PST$L_MAXVALUE];
247    0244 2     STATUS = NMLS_STS_SUC;
248    0245 2
```

```

249      0246 2 ! If the parameter is a string then get the byte count (a byte). If the
250      0247 2 parameter is not a string then get the value of the appropriate width
251      0248 2 (byte, word, longword).
252      0249 2
253      0250 2     IF .SEMANTIC_LIST [PSTSB_FORMAT] EQLU NML$K_STRING
254      0251 2     THEN
255      0252 2       VAL = .LEN
256      0253 2     ELSE
257      0254 2       VAL = .(.ADR)<0,.LEN*8>;
258      0255 2
259      0256 2     Check the minimum parameter value.
260      0257 2
261      0258 2     IF .VAL LSSU .MIN
262      0259 2     THEN
263      0260 2       STATUS = NMLS_STS_PVA;
264      0261 2
265      0262 2     If the maximum value has a zero in it then don't bother to check it.
266      0263 2
267      0264 2     IF .MAX NEQU 0
268      0265 2     AND .VAL GTRU .MAX
269      0266 2     THEN
270      0267 2       STATUS = NMLS_STS_PVA;
271      0268 2
272      0269 2     If the parameter is not within range then signal a parameter value error.
273      0270 2
274      0271 2     IF NOT .STATUS
275      0272 2     THEN
276      0273 2       NML$ERROR_2 (NMASC_STS_PVA, .SEMANTIC_LIST [PSTS_W_DATAID]);
277      0274 2
278      0275 2     RETURN NMLS_STS_SUC
279      0276 2
280      0277 1     END:                                ! End of NML$CHKPRMVAL

```

					003C 00000	.ENTRY	NML\$CHKPRMVAL, Save R2,R3,R4,R5	0190
				50	04 AC 00000000G0040 0A C5 00002	MULL3	#10, CPT_INDEX, R0	0235
				50	9E 3C 0000E	PUSHAB	NML\$AB_CPTABLE[R0]	0237
				50	10 C4 00011	MOVZWL	a(SP)+- R0	
				51	00000000G0040 9E 00014	MULL2	#16, R0	
				55	04 A1 D0 0001C	MOVAB	NML\$AB_PRMSEM[R0], R1	
				54	08 A1 D0 00020	MOVL	4(R1), MIN	0242
				53	01 D0 00024	MOVL	8(R1), MAX	0243
				03	02 A1 91 00027	MOVL	#1, STATUS	0244
				03	06 12 0002B	CMPB	2(R1), #3	0250
				52	08 AC D0 0002D	BNEQ	1S	
				52	08 11 00031	MOVL	LEN, VAL	0252
					08	BRB	2S	
					03 78 00033 1\$:	ASHL	#3, LEN, R0	0254
			52	00 BC 08 AC	50 00 EF 00038	EXTZV	#0, R0, @ADR, VAL	0258
			55		55 52 D1 0003E 2\$:	CMPL	VAL, MIN	
			53		03 1E 00041	BGEQU	3S	0260
					20 CE 00043	MNEGGL	#32, STATUS	0264
					54 D5 00046 3\$:	TSTL	MAX	
					08 13 00048	BEQL	4S	

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$CHKPRMVAL Check parameter value

J 12
16-Sep-1984 00:26:09 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:16 [NML.SRC]NMLPMANIP.B32;1

Page 9
(4)

54	52	D1	0004A	CMPL	VAL, MAX	: 0265	
	03	1B	0004D	BLEQU	4\$		
53	20	CE	0004F	MNEGL	#32, STATUS	: 0267	
0D	53	E8	00052	4\$:	BLBS	STATUS, 5\$: 0271
7E	61	3C	00055	MOVZWL	(R1), -(SP)	: 0273	
7E	10	CE	00058	MNEGL	#16, -(SP)		
00000000G	00	02	FB	0005B	CALLS	#2, NML\$ERROR_2	
50	01	D0	00062	5\$:	MOVL	#1, R0	: 0275
		04	00065		RET		: 0277

; Routine Size: 102 bytes, Routine Base: \$CODE\$ + 008B

```
282    0278 1 %SBTTL 'NML$BLDSETQBF Build SET/CLEAR QIO buffers'
283    0279 1 GLOBAL ROUTINE NML$BLDSETQBF (FUNC, DB,
284    0280 1           KEYID1, KEYLEN1, KEYADR1,
285    0281 1           KEYID2, KEYLEN2, KEYADR2,
286    0282 1           NFBBFDSC, NFBDESC,
287    0283 1           P2BFDSC, P2DSC,
288    0284 1           VALBFDSC, VALDSC)
289    0285 1           : NOVALUE =
290    0286 1
291    0287 1 ++ FUNCTIONAL DESCRIPTION:
292    0288 1 This routine builds a QIO buffer for parameter modifications.
293    0289 1
294    0290 1 FORMAL PARAMETERS:
295    0291 1
296    0292 1
297    0293 1
298    0294 1     FUNC      Control function - NFBSC_FC_SET or NFBSC_FC_CLEAR
299    0295 1     DB        Database ID
300    0296 1     KEYID1   Search key one ID
301    0297 1     KEYLEN1  Search key one length
302    0298 1     KEYADR1  Search key one address
303    0299 1     KEYID2   Search key two ID
304    0300 1     KEYLEN2  Search key two length
305    0301 1     KEYADR2  Search key two address
306    0302 1     NFBBFDSC Descriptor of control function buffer (P1)
307    0303 1     NFBDESC  Descriptor of resulting control function buffer (P1)
308    0304 1     P2BFDSC  Descriptor of P2 buffer
309    0305 1     P2DSC    Descriptor of resulting P2 buffer
310    0306 1     VALBFDSC Descriptor of parameter value buffer (P4)
311    0307 1     VALDSC   Descriptor of resulting parameter value data (P4)
312    0308 1
313    0309 1     NMLSAW_PRM DES List of parameter descriptors
314    0310 1     NMLSGW_PRMDESCNT Count of parameter descriptors
315    0311 1
316    0312 1     OUTPUTS:
317    0313 1
318    0314 1     Output buffers and descriptors built.
319    0315 1     --
320    0316 1
321    0317 2 BEGIN
322    0318 2
323    0319 2 MAP
324    0320 2     NFBBFDSC : REF DESCRIPTOR,
325    0321 2     NFBDESC : REF DESCRIPTOR,
326    0322 2     P2BFDSC : REF DESCRIPTOR,
327    0323 2     P2DSC : REF DESCRIPTOR,
328    0324 2     VALBFDSC : REF DESCRIPTOR,
329    0325 2     VALDSC : REF DESCRIPTOR;
330    0326 2
331    0327 2 LOCAL
332    0328 2     MSGSIZE,
333    0329 2     CPT_INDEX,
334    0330 2     NFB : REF BBLOCK [NFBSC_LENGTH],
335    0331 2     VALADR,
336    0332 2     VALPTR,
337    0333 2     Vallen,
338    0334 2     VALtyp,
```

```
339      0335 2   CPT:      REF BBLOCK [CPT$K_ENTRYLEN];
340      0336 2   PST:      REF BBLOCK [PST$K_ENTRYLEN];
341      0337 2
342      0338 2 BIND
343      0339 2   VALBUF = VALbfdsc [DSCSA_POINTER] : REF BBLOCK;
344      0340 2
345      0341 2
346      0342 2
347      0343 2 | Setup NFB descriptor and header
348      0344 2
349      0345 2
350      0346 2   NFBdesc [DSCSA_POINTER] = NFB = .NFBBFDSC [DSCSA_POINTER];
351      0347 2   CH$FILL(0, $BYTEOFFSET(NFB$L_FLDID), .NFB); ! Zero NFB header
352      0348 2   NFB [NFB$B_FCT] = .FUNC;
353      0349 2   NFB [NFB$B_DATABASE] = .DB;
354      0350 2   NFB [NFB$L_SRCH KEY] = .KEYID1;
355      0351 2   NFB [NFB$L_SRCH2 KEY] = .KEYID2;
356      0352 2   NFB = NFB[NFB$L_FLDID];
357      0353 2
358      0354 2
359      0355 2 | Build the P2 buffer. The P2 buffer contains the values of the search
360      0356 2 keys. The search keys are used by NETACP to identify the database entry
361      0357 2 to be updated.
362      0358 2
363      0359 2 SELECTONEU .KEYID1 OF
364      0360 2   SET
365      0361 2     [NFB$C_EFI_SNIN]: ! Logging filters (sink node)
366      0362 2     NM[SBLDP2 (0, .(.KEYADR1)<0,16>, -1, 0, .P2bfdsc, .P2dsc);
367      0363 2
368      0364 2     [NFB$C_ESI_SNK]: ! Logging sink
369      0365 2     NM[SBLDP2 (0, .(.KEYADR1)<0,8>, -1, 0, .P2bfdsc, .P2dsc);
370      0366 2
371      0367 2     [NFB$C_NDI_ADD]: ! Node (by address)
372      0368 2     NM[SBLDP2 (0, .(.KEYADR1)<0,16>, -1, 0, .P2bfdsc, .P2dsc);
373      0369 2
374      0370 2     [NFB$C_XGI_GRP]: ! Protocol Groups.
375      0371 2
376      0372 2     | Protocol Group is always qualified with a DTE ID. The DTE ID
377      0373 2     | is the second search key.
378      0374 2
379      0375 2     NML$BLDP2 (.KEYLEN1, .KEYADR1, .KEYLEN2, .KEYADR2,
380      0376 2             .P2bfdsc, .P2dsc);
381      0377 2
382      0378 2     [NFB$C_XNI_NET]:
383      0379 3     BEGIN
384      0380 3
385      0381 3     | If search key 1 length is zero, then the QIO is for the "active
386      0382 3     | X-25 Protocol network". Pass NML$BLDP2 the code to indicate that
387      0383 3     | the key value is a word.
388      0384 3
389      0385 3     IF .KEYLEN1 EQL 0 THEN
390      0386 3         KEYLEN1 = -2;
391      0387 3     NML$BLDP2 (.KEYLEN1, .KEYADR1, -1, 0, .P2bfdsc, .P2dsc);
392      0388 2     END;
393      0389 2
394      0390 2     [OTHERWISE]:
395      0391 2     NML$BLDP2 (.KEYLEN1, .KEYADR1, -1, 0, .P2bfdsc, .P2dsc);
```

```
396      0392 2
397      0393 2      TES;
398      0394 2
399      0395 2
400      0396 2      ! Setup parameter value buffer descriptor
401      0397 2
402      0398 2
403      0399 2      VALDSC [DSC$A_POINTER] = VALPTR = .VALBFDSC [DSC$A_POINTER];
404      0400 2
405      0401 2
406      0402 2      For each entry in the parameter descriptor list, add its ACP identifier
407      0403 2      to the NFB and its value to the value buffer.
408      0404 2
409      0405 2      INCR I FROM 0 TO .NML$GW_PRMDESCNT - 1 DO
410      0406 3      BEGIN
411      0407 3
412      0408 3      CPT_INDEX = .NMLSAW_PRM DES [.I, PDB$W_INDEX];
413      0409 3      CPT = NML$AB_CPTABLE [.CPT_INDEX, 0,0,0,0];
414      0410 3      PST = NML$AB_PRMSEM [.CPT [CPTSW_PSTINDEX], 0,0,0,0];
415      0411 3      Vallen = .NMLSAW_PRM DES [.I, PDB$W_COUNT];
416      0412 3      VALADR = .NMLSAW_PRM DES [.I, PDB$A_POINTER];
417      0413 3
418      0414 4      IF (.VALPTR + Vallen + 2 LSSU
419      0415 3          .VALBFDSC [DSC$A_POINTER] + .VALBFDSC [DSC$W_LENGTH]) AND
420      0416 4          (.NFB + 4 LSSU
421      0417 4          .NFBBFDSC [DSC$A_POINTER] + .NFBBFDSC [DSC$W_LENGTH])
422      0418 4
423      0419 3      THEN
424      0420 4      BEGIN
425      0421 4          NFB[0,0,32,0] = .PST [PST$L_NFBID];
426      0422 4          NFB = .NFB + 4;
427      0423 4
428      0424 4      IF .Vallen GTRU 0
429      0425 4      THEN
430      0426 5      BEGIN
431      0427 6          VALTYP = .(PST [PST$L_NFBID])
432      0428 5              <$BITPOSITION (NFB$V_TYP),
433      0429 5                  $FIELDWIDTH (NFB$V_TYP)>;
434      0430 5      IF .VALTYP EQLU NFB$C_TYP_STR
435      0431 5      THEN
436      0432 6      BEGIN
437      0433 6
438      0434 6          (.VALPTR)<0,16> = .Vallen;      ! Set count
439      0435 6          VALPTR = .VALPTR + 2;
440      0436 6          VALPTR = CH$MOVE (.Vallen, .VALADR, .VALPTR);
441      0437 6
442      0438 6      END
443      0439 5      ELSE
444      0440 6      BEGIN
445      0441 6
446      0442 6          (.VALPTR)<0,32> = .(.VALADR)<0,.Vallen*8>;
447      0443 6          VALPTR = .VALPTR + 4;      ! Increment data pointer
448      0444 6
449      0445 5      END;
450      0446 4      END;
451      0447 4      END;
452      0448 3      ELSE
```

```

453 0449 4 BEGIN
454 0450 4
455 0451 4 NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSBSM MSG FLD; ! Set message text flag
456 0452 4 NML$AB_MSGBLOCK [MSB$B_CODE] = NMASC STS MPR;
457 0453 4 NML$AB_MSGBLOCK [MSB$L_TEXT] = NML$ QIOBFOVF;
458 0454 4 NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE); ! Build message
459 0455 4 SSIGNAL_MSG (NML$AB_SNDBUFFER, .MSGSIZE); ! Signal it
460 0456 4
461 0457 3 END;
462 0458 3
463 0459 2 END;
464 0460 2
465 0461 2 NFB [0,0,32,0] = NFBSC_ENDOFLIST;
466 0462 2 NFBDS [DSC$W_LENGTH] = .NFB - .NFBDS[DSC$A_POINTER] + 4;
467 0463 2
468 0464 2 VALDSC [DSC$W_LENGTH] = .VALPTR - .VALDSC[DSC$A_POINTER];
469 0465 2
470 0466 1 END; ! End of NML$BLDSETQBF

```

				OFFC 00000	.ENTRY	NML\$BLDSETQBF, Save R2,R3,R4,R5,R6,R7,R8,-	0279
		5E	34	14 C2 00002	SUBL2	R9 R10, R11	
	50	24 AC		04 DD 00005	PUSHL	#20, SP	0339
	51	28 AC		60 D0 0000D	ADDL3	#4, NFBFDSC, R0	0346
10	00	61 6E		04 C1 00010	MOVL	(R0), NFB	
		56		56 D0 00015	ADDL3	#4, NFBDS, R1	
		61		00 2C 00018	MOVL	NFB, (R1)	
		6E		66 0001D	MOVCS	#0, (SP), #0, #16, (NFB)	0347
		86	04	AC 90 0001E	MOVB	FUNC, (NFB)+	0348
	01	A6	08	AC 90 00022	MOVB	DB, 1(NFB)	0349
		50	0C	AC D0 00027	MOVL	KEYID1, R0	0350
	03	A6	18	50 D0 0002B	MOVL	R0, 3(NFB)	
	07	A6	18	AC D0 0002F	MOVL	KEYID2, 7(NFB)	0351
	06010010	56 8F		OF CO 00034	ADDL2	#15, NFB	0352
		8F		50 D1 00037	CMPL	R0, #100728848	0361
	07010010	8F		21 13 0003E	BEQL	2\$	
		8F		50 D1 00040	CMPL	R0, #117506064	0364
		7E	2C	OF 12 00047	BNEQ	1\$	
		7E	2C	AC 7D 00049	MOVQ	P2BFDSC, -(SP)	0365
		7E	14	7E D4 0004D	CLRL	-(SP)	
		7E	14	01 CE 0004F	MNEG	#1, -(SP)	
		7E	14	BC 9A 00052	MOVZBL	@KEYADR1, -(SP)	
	02010012	8F		16 11 00056	BRB	3\$	
		8F		50 D1 00058	CMPL	R0, #33619986	0367
		7E	2C	11 12 0005F	BNEQ	4\$	
		7E	2C	AC 7D 00061	MOVQ	P2BFDSC, -(SP)	0368
		7E	14	7E D4 00065	CLRL	-(SP)	
		7E	14	01 CE 00067	MNEG	#1, -(SP)	
		7E	14	BC 3C 0006A	MOVZWL	@KEYADR1, -(SP)	
	0A020041	8F		32 11 00070	CLRL	-(SP)	
		8F		50 D1 00072	BRB	8\$	
		8F		50 D1 00072	CMPL	R0, #167903297	0370

NML\$PMANIP
V04-000

NML internal parameter manipulation module NML\$BLDSETQBF Build SET/CLEAR QIO buffers

B 13
16-Sep-1984 00:26:01
14-Sep-1984 12:50:10

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32:1

Page 14
(5)

★ ★ |

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$BLDSETQBF Build SET/CLEAR QIO buffers

C 13
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 15
(5)

00000000G	00		3F	11	0015F	BRB	12\$: 0424
00000000G	00		04	D0	00161	11\$:	MOVL #4, NML\$AB_MSGBLOCK	: 0451
00000000G	00	00000000G	05	8E	00168	MNEG8	#5, NML\$AB_MSGBLOCK+4	: 0452
00000000G	00	00000000G	8F	D0	0016F	MOVI.	#NMLS_QIOBF0VF, NML\$AB_MSGBLOCK+12	: 0453
			14	AE	9F	PUSHAB	MSGSIZE	: 0454
			00000000G	00	0017A	PUSHAB	NML\$AB_MSGBLOCK	
00000000G	00		02	FB	00183	CALLS	#2, NM\$BLD_REPLY	: 0455
			14	AE	DD	PUSHL	MSGSIZE	
			00000000G	00	0018A	PUSHAB	NML\$AB_SNDBUFFER	
			01F90000	8F	DD	PUSHL	#33095680	
02	00000000G	00	03	FB	00199	CALLS	#3, LIB\$SIGNAL	
	5A	OC	AE	F2	001A0	AOBLSS	12(SP), I, 13\$: 0405
			03	11	001A5	BRB	14\$	
			FF1E	31	001A7	13\$:	BRW	9\$
				66	D4	14\$:	CLRL	(NFB)
50	28	AC		04	C1	001AC	ADDL3	#4, NFBDSC, R0
		56		60	C2	001B1	SUBL2	(R0), R6
28	BC	56		04	A1	001B4	ADDW3	#4, R6, @NFBDSC
	6B	53	04	AB	A3	001B9	SUBW3	4(R11), VALPTR, (R11)
				04	001BE		RET	: 0464
								: 0466

; Routine Size: 447 bytes, Routine Base: \$CODE\$ + 00F1

; 471 0467 1

```
473      0468 1 %SBTTL 'NML$ADD_FIELDS Add parameter fields to record'  
474      0469 1 GLOBAL ROUTINE NMLSADD_FIELDS (BUFSIZE, RTNDSC) =  
475      0470 1  
476      0471 1 ++  
477      0472 1 FUNCTIONAL DESCRIPTION:  
478      0473 1  
479      0474 1 This routine adds fields to a permanent data base record.  
480      0475 1  
481      0476 1 FORMAL PARAMETERS:  
482      0477 1  
483      0478 1 BUFSIZE Maximum size of the record buffer.  
484      0479 1 RTNDSC Address of the current record descriptor.  
485      0480 1  
486      0481 1 IMPLICIT INPUTS:  
487      0482 1  
488      0483 1  
489      0484 1  
490      0485 1  
491      0486 1  
492      0487 1  
493      0488 1  
494      0489 1  
495      0490 1  
496      0491 1  
497      0492 1  
498      0493 1  
499      0494 1  
500      0495 1  
501      0496 1  
502      0497 1  
503      0498 1  
504      0499 1 --  
505      0500 1  
506      0501 2 BEGIN  
507      0502 2  
508      0503 2 LOCAL  
509      0504 2 CPT_INDEX, | Change parameter table index  
510      0505 2 SEM_INDEX, | Semantic table index  
511      0506 2 FLDEEN, | Field length  
512      0507 2 FLDADR, | Field address  
513      0508 2 MSGSIZE, | Message size  
514      0509 2 ROUTINE_ADDR, | Temporary routine address  
515      0510 2 STATUS;  
516      0511 2  
517      0512 2 INCR I FROM 0 TO .NML$GW_PRMDESCNT - 1 DO  
518      0513 3 BEGIN  
519      0514 3  
520      0515 3 FLDLEN = .NMLSAW_PRM DES [.I, PDB$W COUNT];  
521      0516 3 FLDADR = .NMLSAW_PRM DES [.I, PDB$A_POINTER];  
522      0517 3  
523      0518 3 CPT_INDEX = .NMLSAW_PRM DES [.I, PDB$W INDEX];  
524      0519 3 ROUTINE_ADR = .NMLSAB_CPTABLE [.CPT_INDEX, CPT$A_DEFINE RTN];  
525      0520 3 SEM_INDEX = .NMLSAB_CPTABLE [.CPT_INDEX, CPT$W_P$TINDEX];  
526      0521 3  
527      0522 4 IF NOT (STATUS =  
528      0523 4 (.ROUTINE_ADR) (NMLSAB_PRMSEM [.SEM_INDEX, 0,0,0,0],  
529      0524 4 .BUFSIZE,
```

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$ADD_FIELDS Add parameter fields to record

E 13
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 17
(6)

```
530 0525 4 .FLDLEN,  
531 0526 4 .FLDADR,  
532 0527 4 .RTNDSC})  
533 0528 3 THEN RETURN .STATUS  
534 0529 3  
535 0530 3  
536 0531 2 END;  
537 0532 2  
538 0533 2 RETURN NML$_STS_SUC  
539 0534 2  
540 0535 1 END; ! End of NML$ADD_FIELDS
```

		03FC 00000	.ENTRY	NML\$ADD_FIELDS, Save R2,R3,R4,R5,R6,R7,R8,- : 0469	
		59 00000000G 00 9E 00002	MOVAB	NMLSAW_PRM DES+2, R9	
		55 00000000G 00 3C 00009	MOVZWL	NMLSGW_PRMDESCNT, R5	0512
		54 01 CE 00010	MNEGL	#1, I	0524
		47 11 00013	BRB	2S	
		6944 7F 00015 1\$:	PUSHAQ	NMLSAW_PRM DES+2[I]	0515
		58 9E 3C 00018	MOVZWL	a(SP)+, FLDLEN	
		57 02 A944 7F 0001B	PUSHAQ	NMLSAW_PRM DES+4[I]	0516
		57 9E D0 0001F	MOVL	a(SP)+, FLDADR	
		FE A944 7F 00022	PUSHAQ	NMLSAW_PRM DES[I]	0518
		53 9E 3C 00026	MOVZWL	a(SP)+, CPT INDEX	
		53 0A C5 00029	MULL3	#10, CPT INDEX, R1	0519
51		00000000G0041 9F 0002D	PUSHAB	NML\$AB_CPTABLE+2[R1]	
		56 9E D0 00034	MOVL	a(SP)+, ROUTINE ADR	0520
		56 00000000G0041 9F 00037	PUSHAB	NML\$AB_CPTABLE[R1]	
		52 9E 3C 0003E	MOVZWL	a(SP)+, SEM_INDEX	
		08 AC DD 00041	PUSHL	RTNDSC	0527
		57 DD 00044	PUSHL	FLDADR	0526
		58 DD 00046	PUSHL	FLDLEN	0525
		04 AC DD 00048	PUSHL	BUFSIZE	0524
51		52 04 78 0004B	ASHL	#4, SEM_INDEX, R1	0523
		00000000G0041 9F 0004F	PUSHAB	NML\$AB_PRMSEM[R1]	
		66 05 FB 00056	CALLS	#5, (ROUTINE_ADR)	
		07 50 E9 00059	BLBC	STATUS, 3\$	
B5		54 55 F2 0005C 2\$:	A0BLSS	R5, I, 1\$	0522
		50 01 D0 00060	MOVL	#1, R6	0533
		04 00063 3\$:	RET		0535

: Routine Size: 100 bytes, Routine Base: \$CODE\$ + 02B0

542 0536 1 %SBTTL 'NML\$DEL_FIELDS Delete parameter fields from record'
543 0537 1 GLOBAL ROUTINE NML\$DEL_FIELDS (RTNDSC) =
544 0538 1
545 0539 1 !++
546 0540 1 FUNCTIONAL DESCRIPTION:
547 0541 1
548 0542 1 This routine deletes the entire list of parameters in the parameter
549 0543 1 descriptor buffer from the specified record buffer.
550 0544 1
551 0545 1 FORMAL PARAMETERS:
552 0546 1
553 0547 1 RTNDSC contains the address of the current record descriptor.
554 0548 1
555 0549 1 IMPLICIT INPUTS:
556 0550 1
557 0551 1 NML\$GW_PRMDESCNT contains the number of parameter descriptors.
558 0552 1 NML\$AW_PRM DES is a list of parameter descriptors.
559 0553 1 NML\$AB_PRMSEM is the parameter semantic table.
560 0554 1
561 0555 1 IMPLICIT OUTPUTS:
562 0556 1
563 0557 1 The record descriptor pointed to by RTNDSC is updated to reflect
564 0558 1 any fields deleted from the record.
565 0559 1
566 0560 1 ROUTINE VALUE:
567 0561 1 COMPLETION CODES:
568 0562 1
569 0563 1 Always returns success (NML\$_STS_SUC).
570 0564 1
571 0565 1 SIDE EFFECTS:
572 0566 1
573 0567 1 NONE
574 0568 1
575 0569 1 --
576 0570 1
577 0571 2 BEGIN
578 0572 2
579 0573 2 LOCAL
580 0574 2 CPT_INDEX,
581 0575 2 SEM_INDEX,
582 0576 2 ROUTINE_ADR;
583 0577 2
584 0578 2 INCR I FROM 0 TO .NML\$GW_PRMDESCNT - 1 DO
585 0579 3 BEGIN
586 0580 3
587 0581 3 CPT_INDEX = .NML\$AW_PRM DES [.I, PDBSW_INDEX];
588 0582 3 ROUTINE_ADR = .NML\$AB_CPTABLE [.CPT_INDEX, CPTSA_PURGE RTN];
589 0583 3 SEM_INDEX = .NML\$AB_CPTABLE [.CPT_INDEX, CPTSW_PSTINDEX];
590 0584 3
591 0585 3 (.ROUTINE_ADR) (.RTNDSC,
592 0586 3 NML\$AB_PRMSEM [.SEM_INDEX, 0,0,0,0]);
593 0587 3
594 0588 2 END;
595 0589 2
596 0590 2 RETURN NML\$_STS_SUC
597 0591 2
598 0592 1 END;

! End of NML\$DEL_FIELDS

		007C 00000	.ENTRY NMLSDEL_FIELDS, Save R2,R3,R4,R5,R6	: 0537
		55 0000000G 00 3C 00002	MOVZWL NMLSGW_PRMDESCNT, R5	: 0578
		54 01 CE 00009	MNEG L #1, I	: 0585
		33 11 0000C	BRB 2\$	
		00000000G0044 7F J000E 1\$:	PUSHAQ NMLSAW_PRM DES[I]	: 0581
50	53	9E 3C 00015	MOVZWL @(SP)+, CPT INDEX	: 0582
	53	0A C5 00018	MULL3 #10, CPT INDEX, R0	: 0583
	56	00000000G0040 9F 0001C	PUSHAB NMLSAB_CPTABLE+6[R0]	: 0584
	56	9E D0 00023	MOVL @(SP)+, ROUTINE ADR	: 0585
	52	00000000G0040 9F 00026	PUSHAB NMLSAB_CPTABLE[R0]	: 0586
50	52	9E 3C 0002D	MOVZWL @(SP)+, SEM INDEX	: 0587
	52	04 78 00030	ASHL #4, SEM INDEX, R0	: 0588
		00000000G0040 9F 00034	PUSHAB NMLSAB_PRMSEM[R0]	: 0589
		04 AC DD 0003B	PUSHL RTNDSC	: 0590
C9	66	02 FB 0003E	CALLS #2, (ROUTINE_ADR)	: 0591
	54	55 F2 00041 2\$:	A0BLSS R5, I, 1\$: 0592
	50	01 D0 00045	MOVL #1, R0	
		04 00048	RET	

; Routine Size: 73 bytes, Routine Base: \$CODE\$ + 0314

```
600      0593 1 %SBTTL 'NMLSREADPARLIST Show parameters from buffer'
601      0594 1 GLOBAL ROUTINE NMLSREADPARLIST(BUFDSC, MSGSIZE, TABDSC, DATDSC) =
602      0595 1
603      0596 1 ++
604      0597 1 FUNCTIONAL DESCRIPTION:
605      0598 1
606      0599 1 This routine builds a message from the list of parameters specified.
607      0600 1
608      0601 1 FORMAL PARAMETERS:
609      0602 1
610      0603 1     BUFDSC      Address of message buffer descriptor.
611      0604 1     MSGSIZE     Address of longword to contain resulting message size.
612      0605 1     TABDSC      Address of parameter table descriptor.
613      0606 1     DATDSC      Address of data buffer descriptor.
614      0607 1
615      0608 1 IMPLICIT INPUTS:
616      0609 1     NONE
617      0610 1
618      0611 1 IMPLICIT OUTPUTS:
619      0612 1     NONE
620      0613 1
621      0614 1
622      0615 1
623      0616 1 ROUTINE VALUE:
624      0617 1 COMPLETION CODES:
625      0618 1
626      0619 1     Always returns success (NML$STS_SUC).
627      0620 1
628      0621 1 SIDE EFFECTS:
629      0622 1     NONE
630      0623 1
631      0624 1
632      0625 1 --
633      0626 1
634      0627 2 BEGIN
635      0628 2
636      0629 2 MAP
637      0630 2     TABDSC : REF DESCRIPTOR;
638      0631 2
639      0632 2 LOCAL
640      0633 2     INDEX;
641      0634 2
642      0635 2 BIND
643      0636 2     TABLE = TABDSC [DSC$A_POINTER] : REF BBLOCKVECTOR [, 6];
644      0637 2
645      0638 2
646      0639 2 If table address is null then the specified information type is not
647      0640 2 applicable to this entity.
648      0641 2
649      0642 2 IF .TABLE EQLA 0
650      0643 2 THEN
651      0644 2     NML$ERROR_1 (NMASCSTS_FOP);
652      0645 2
653      0646 2 INCR I FROM 0 TO .TABDSC [DSC$W_LENGTH] - 1 DO
654      0647 2     BEGIN
655      0648 2
656      0649 3     INDEX = .TABLE [.I, 0,0,16,0]; ! Get table index
```

```

657 0650 3
658 0651 3
659 0652 3
660 0653 3
661 0654 3
662 0655 3
663 0656 2
664 0657 2
665 0658 2
666 0659 2
667 0660 1

(.TABLE [.I, 2,0,32,0]) (NMLSAB PRMSEM [.INDEX, 0,0,0,0],
    .BUFDSC,
    .MSGSIZE,
    .DATDSC);

END;

RETURN NMLS_STS_SUC

END;

```

! End of NMLSREADPARLIST

			003C 00000	.ENTRY	NMLSREADPARLIST, Save R2,R3,R4,R5	0594
54	OC AC		04 C1 00002	ADDL3	#4, TABDSC, R4	0636
			64 D5 00007	TSTL	(R4)	0642
			0A 12 00009	BNEQ	1\$	0644
		00000000G 7E 00	0D CE 0000B	MNEGL	#13, -(SP)	0644
		0C 55 52	01 FB 0000E	CALLS	#1, NMLSERROR_1	0646
			BC 3C 00015	MOVZWL	@TBDSC, R5	0646
			01 CE 00019	MNEGL	#1, I	0652
			20 11 0001C	BRB	3\$	0652
51	52		06 C5 0001E	2\$: MULL3	#6, I, R1	0649
	51		64 C0 00022	ADDL2	(R4), R1	0652
	53		61 3C 00025	MOVZWL	(R1), INDEX	0654
		10	AC DD 00028	PUSHL	DATDSC	0654
50	7E 04	00000000G0040	AC 7D 0002B	MOVQ	BUFDSC, -(SP)	0652
	53		04 78 0002F	ASHL	#4, INDEX, R0	0651
	02 B1		9F 00033	PUSHAB	NMLSAB PRMSEM[R0]	0651
	52		04 FB 0003A	CALLS	#4, @27R1)	0646
DC	50		55 F2 0003E	A0BLSS	R5, I, 2\$	0658
			01 D0 00042	MOVL	#1, R0	0660
			04 00045	RET		

: Routine Size: 70 bytes, Routine Base: \$CODE\$ + 035D

```
669      0661 1 %SBTTL 'NML$SHOWPARLIST Show parameters from QIO buffer'
670      0662 1 GLOBAL ROUTINE NML$SHOWPARLIST (BUFDSC, MSGSIZE, TABDSC, DATDSC, DATPTR) =
671      0663 1
672      0664 1 ++ FUNCTIONAL DESCRIPTION:
673      0665 1
674      0666 1 This routine builds a message from the list of parameters specified.
675      0667 1
676      0668 1 FORMAL PARAMETERS:
677      0669 1
678      0670 1
679      0671 1     BUFDSC      Address of message buffer descriptor.
680      0672 1     MSGSIZE     Address of longword to contain resulting message size.
681      0673 1     TABDSC      Address of parameter table descriptor.
682      0674 1     DATDSC      Address of data buffer descriptor.
683      0675 1     DATPTR      Address of data buffer pointer.
684      0676 1
685      0677 1     IMPLICIT INPUTS:
686      0678 1         NONE
687      0679 1
688      0680 1     IMPLICIT OUTPUTS:
689      0681 1         NONE
690      0682 1
691      0683 1     ROUTINE VALUE:
692      0684 1     COMPLETION CODES:
693      0685 1         Always returns success (NML$_STS_SUC).
694      0686 1
695      0687 1     SIDE EFFECTS:
696      0688 1         NONE
697      0689 1
698      0690 1     0691 1
699      0692 1         NONE
700      0693 1
701      0694 1         --
702      0695 1
703      0696 2     BEGIN
704      0697 2
705      0698 2     MAP
706      0699 2         TABDSC : REF DESCRIPTOR;
707      0700 2
708      0701 2     LOCAL
709      0702 2         INDEX;
710      0703 2
711      0704 2     BIND
712      0705 2         TABLE = TABDSC [DSC$A_POINTER] : REF BBLOCKVECTOR [, 6];
713      0706 2
714      0707 2     INCR I FROM 0 TO .TABDSC [DSC$W_LENGTH] - 1 DO
715      0708 2         BEGIN
716      0709 3         INDEX = .TABLE [.I, 0,0,16,0]; ! Get table index
717      0710 3
718      0711 3
719      0712 3     (.TABLE [.I, 2,0,32,0]) (NML$AB_PRMSEM [.INDEX, 0,0,0,0],
720      0713 3         .BUFDSC,
721      0714 3         .MSGSIZE,
722      0715 3         .DATDSC,
723      0716 3         .DATPTR);
```

```

: 726    0718 2      END;
: 727    0719 2
: 728    0720 2      RETURN NML$_STS_SUC
: 729    0721 2
: 730    0722 1      END;

```

! End of NML\$SHOWPARLIST

				003C 00000	.ENTRY	NML\$SHOWPARLIST, Save R2,R3,R4,R5	0662
55	0C	AC	0C	04 C1 00002	ADDL3	#4, TABDSC, R5	0705
		54		BC 3C 00007	MOVZWL	@TBDSC, R4	0707
		52		01 CE 0000B	MNEGL	#1, I	0713
				21 11 0000E	BRB	2S	
51		52		06 C5 00010	1\$: MULL3	#6, I, R1	0710
		51		65 C0 00014	ADDL2	(R5), R1	
		53		61 3C 00017	MOVZWL	(R1), INDEX	
		7E	10	AC 7D 0001A	MOVQ	DATDSC, -(SP)	0715
		7E	04	AC 7D 0001E	MOVQ	BUFDSC, -(SP)	0713
50		53		04 78 00022	ASHL	#4, INDEX, R0	0712
	02	B1	00000000G0040	9F 00026	PUSHAB	NML\$AB PRMSEM[R0]	
		52		05 FB 0002D	CALLS	#5, @2TR1)	
		50		54 F2 00031	AOBLSS	R4, I, 1\$	0707
				01 D0 00035	MOVL	#1, R0	0720
				04 00038	RET		0722

; Routine Size: 57 bytes, Routine Base: \$CODE\$ + 03A3

```
732 0723 1 %SBTTL 'NMLSBLDALLDES Build parameter descriptors from record'
733 0724 1 GLOBAL ROUTINE NMLSBLDALLDES (RECDSC, TABDSC) =
734 0725 1
735 0726 1 ++
736 0727 1 FUNCTIONAL DESCRIPTION:
737 0728 1
738 0729 1 This routine is used by SET ALL functions to build parameter
739 0730 1 descriptors from a permanent data base record.
740 0731 1
741 0732 1 FORMAL PARAMETERS:
742 0733 1
743 0734 1 RECDSC Address of the current record descriptor.
744 0735 1 TABDSC Address of parameter table descriptor.
745 0736 1
746 0737 1 IMPLICIT INPUTS:
747 0738 1 NMLSAB_PRMSEM is the parameter semantic table.
748 0739 1
749 0740 1 IMPLICIT OUTPUTS:
750 0741 1
751 0742 1
752 0743 1
753 0744 1
754 0745 1
755 0746 1
756 0747 1
757 0748 1
758 0749 1
759 0750 1
760 0751 1
761 0752 1
762 0753 1
763 0754 1
764 0755 1
765 0756 2
766 0757 2
767 0758 2
768 0759 2 RECDSC : REF DESCRIPTOR,
769 0760 2 TABDSC : REF DESCRIPTOR;
770 0761 2
771 0762 2
772 0763 2 LOCAL
773 0764 2
774 0765 2
775 0766 2
776 0767 2
777 0768 2
778 0769 2
779 0770 2
780 0771 2
781 0772 2
782 0773 3
783 0774 3
784 0775 3
785 0776 3
786 0777 3
787 0778 3
788 0779 3
-- BEGIN
MAP
    RECDSC : REF DESCRIPTOR,
    TABDSC : REF DESCRIPTOR;
LOCAL
    FLDADR,
    FLDSIZE,
    INDEX;
BIND
    TABLE = TABDSC [DSC$A_POINTER] : REF BLOCK;
NML$GW_PRMDESCNT = 0;           ! Reset parameter descriptor count
INCR I FROM 0 TO .TABDSC [DSC$W_LENGTH] - 1 DO
    BEGIN
        FLDADR = 0;
        IF NMASSEARCHFLD (.RECDSC,
                            TABLE [.I,0,16,0],
                            FLDSIZE,
```

```

789   0780 3          FLDADR)
790   0781 3          THEN BEGIN
791   0782 4
792   0783 4
793   0784 4          INDEX = .TABLE [.I,16,16,0];
794   0785 4
795   0786 4          NML$SAVEPARAM (.INDEX,
796   0787 4          .FLDSIZE,
797   0788 4          .FLDADR);
798   0789 3          END;
799   0790 2          END;
800   0791 2
801   0792 2          RETURN NML$_STS_SUC
802   0793 2
803   0794 1          END;

```

! End of NML\$BLDALLDL

				003C 00000	.ENTRY	NML\$BLDALLDL, Save R2,R3,R4,R5	0724
53	08	5E AC	00000000G	08 C2 00002	SUBL2	#8, SP	0768
				04 C1 00005	ADDL3	#4, TABDSC, R3	0770
		54 08		00 B4 0000A	CLRW	NML\$GW PRMDESCNT	0772
		52		BC 3C 00010	MOVZWL	@TABDSC, R4	0777
				01 CE 00014	MNEGL	#1, I	
				30 11 00017	BRB	2\$	
				6E D4 00019	CLRL	FLDADR	0775
				1\$: 5E DD 0001B	PUSHL	SP	0777
				08 AE 9F 0001D	PUSHAB	FLDSIZE	
				00 B342 DF 00020	PUSHAL	@(R3)[I]	0778
		7E		9E 3C 00024	MOVZWL	@(SP)+, -(SP)	
				04 AC DD 00027	PUSHL	RECDSC	0777
			00000000G	00 04 FB 0002A	CALLS	#4, NMASSEARCHFLD	
				15 50 E9 00031	BLBC	R0, 2\$	
				00 B342 DF 00034	PUSHAL	@(R3)[I]	0784
55	9E	10		10 EF 00038	EXTZV	#16, #16, @(SP)+, INDEX	0788
				6E DD 0003D	PUSHL	FLDADR	0787
				08 AE DD 0003F	PUSHL	FLDSIZE	0786
		FBDB CF		55 DD 00042	PUSHL	INDEX	
	CC	52		03 FB 00044	CALLS	#3, NML\$SAVEPARAM	
		50		54 F2 00049	AOBLS	R4, I 1\$	0772
				01 D0 0004D	MOVL	#1, R0	0792
				04 00050	RET		0794

: Routine Size: 81 bytes, Routine Base: \$CODE\$ + 03DC

NMLSPMANIP
V04-000

NML internal parameter manipulation module
NMLSBLDALLDES Build parameter descriptors from

N 13
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 26
(11)

805 0795 1 END
806 0796 1
807 0797 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	1069	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols	Pages	Processing
	Total	Loaded	Mapped	Time
\$255\$DUA28:[NML.OBJ]NMLLIB.L32;1	341	38	11	00:00.1
\$255\$DUA28:[SHRLIB]NMALIBRY.L32;1	887	3	0	00:00.2
\$255\$DUA28:[SHRLIB]NET.L32;1	1279	14	1	00:00.3
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	5	0	00:03.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:NMLPMANIP/OBJ=OBJ\$:NMLPMANIP MSRC\$:NMLPMANIP/UPDATE=(ENH\$:NMLPMANIP)

Size: 1069 code + 0 data bytes
Run Time: 00:24.0
Elapsed Time: 01:02.4
Lines/CPU Min: 1991
Lexemes/CPU-Min: 12432
Memory Used: 169 pages
Compilation Complete

0285 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

